**Parte 1**

Primeiramente, abri as pastas e iniciei a primeira parte do trabalho: alterar o conteúdo delas. Minha dupla e eu escolhemos desenvolver o conteúdo do site voltado para o álbum *Ruby*, da artista coreana Jennie, com o objetivo de criar um informativo.

Comecei alterando as informações do arquivo **index.html**, a página principal, para depois modificar as demais páginas. Adicionei as imagens relacionadas ao tema do projeto na pasta **images** e instalei o **nodemon**, para que o servidor reinicie automaticamente após cada alteração, tornando o processo de desenvolvimento mais dinâmico.Mudei os gets do arquvio main.js para poder utilizar as imagens novas que eu adicionei, certa dificuldade inicial em lidar com as imagens do projeto para aparecerem no site, tivemos que parar e entender o codigo presente no arquivo para ver como ele usa requisições e a função getfile presente no arquvio utils para buscar a imagem e coloca-la no site, como não mudamos a logica do site optamos por manter os nomes dos arquivos (mesmo que tenhamos mudado o conteudo das paginas) deixamos tambem as imagens que já tinham no projeto na pasta, mesmo que não utilizando elas, só trocamos os gets para buscar as novas imagens

**Parte 2**

A pasta **public** contém:

* A subpasta **css**, com as estilizações das páginas;
* A subpasta **images**, com as imagens utilizadas no site.

A pasta **views** contém as páginas e seus respectivos arquivos HTML.

### O arquivo main

O código no arquivo main cria um **servidor**. O servidor responde a diferentes requisições HTTP, como abrir páginas HTML, carregar imagens, arquivos CSS e JavaScript.

O servidor é criado com o módulo nativo http e configurado para ouvir na **porta 3000**.

O código utiliza módulos personalizados para organizar:**Rotas**, **Tipos de conteúdo**, **Funções auxiliares**

São definidos caminhos para responder a diferentes URLs, como:

/ → Página inicial (index.html)

/courses.html, /contact.html → Outras páginas

Arquivos de imagem, CSS e JS também são tratados por suas respectivas rotas.

Cada rota envia uma resposta com: Um **código de status HTTP** (como 200 para sucesso)Um **tipo de conteúdo apropriado** (HTML, CSS, imagem etc.)O conteúdo do arquivo solicitado (usando utils.getFile)

**Arquivo content types**

**Arquivos package.json**

**Arquivo router**

**Arquivo utils**

**Como os arquivos interagem entre eles**